



برنامه نویسی پیشرفت زمستان و بهار ۱۳۹۸-۹۹ - دانشکده علوم ریاضی دانشگاه صنعتی شریف

با توجه به شرایط خاص پیش آمده تیم درس برنامه نویسی پیشرفت‌نمایه بدون تاثیر در ارزیابی برای بررسی میزان پیشرفت مطالعه برگزار کند. هدف از این پرسشنامه بررسی پیشرفت عملکرد و مطالعه شما است. هدف از این پرسشنامه این موارد است:

- جبران فاصله ایجاد شده میاد دانشجوها با همیگر که امکان ارائه بازخورد پیشرفت مطالعه به یکدیگر را ایجاد می‌کند.
- جبران فاصله ایجاد شده میان دانشجوها و تیم درس برای دریافت بازخورد پیشرفت تحصیلی بازخورد هر دانشجو به خود در رابطه با پیشرفت مناسب در مطالعه و یادگیری مفاهیم از طریق منابع درس جهتدهی به اشکالاتی که شما ممکن هست هنوز در جریان وجود نقطه ضعف خود در این رابطه نباشید و رفع آنها در جلسه‌های رفع اشکال آنلاین
- دقت کنید که این پرسشنامه تنها مرجع برای بازخورد پیشرفت مناسب شما در مطالعه منابع نیست. تمرين‌ها و پژوهه‌های نیز سهم بزرگی در این مساله دارند. پس حتماً پس از پاسخ به سوال‌ها و ارسال پاسخ‌ها، اشکال‌ها و ابهام‌هایی که داشتید در جلسه‌های آنلاین رفع اشکال در میان بگذارید و رفع کنید و از این فرصت استفاده کنید.

توضیحات

- نتیجه این پرسشنامه تاثیری در ارزیابی نهایی این درس ندارد.
- این پرسشنامه برای اطمینان بیشتر از اینکه مسیر درس را درست طی می‌کنید طراحی شده.
 - اگر نیاز به بررسی صحت پیشرفت‌نمایه دارید حتماً در این پرسشنامه شرکت کنید.
 - در صورتی که با مطالب درس به درستی پیش آمده باشید می‌توانید به تمام سوال‌ها پاسخ دهید.
 - در صورتی که به بخشی از هر سوال تسلط ندارید یا احتیاج به بررسی صحت پاسخ‌ها دارید حتماً در جلسه‌های رفع اشکال شرکت کنید و اشکال یا ابهام‌های خود را رفع کنید.
 - سعی کنید جواب‌ها کوتاه و دقیق باشند که مرور جواب در جلسه رفع اشکال سریع‌تر انجام شود.
 - از آنجایی که این پرسشنامه برای یادگیری طراحی شده می‌توانید در پر کردن سوال‌ها با هر فردی مشورت و همفکری کنید.

نحوه انجام پرسشنامه

- برای پاسخ به این پرسشنامه یک نسخه از این فایل را از منو فایل و گزینه گرفتن یک کپی برای خود ایجاد کنید و جواب‌های آن را در همین فایل بنویسید.
- پس از جواب دادن به سوال‌ها آن را در قالب PDF دانلود کنید.
- فایل PDF در یک ریپازیتوری [github](#) بارگذاری کنید.
- آدرس این ریپازیتوری را در یک فایل یک خطی با پسوند `.github` در بخش پرسشنامه بررسی پیشرفت بارگذاری کنید.

سوال‌ها

سوال ۱

خروجی این برنامه را بست بیاورید و به ازای هر خط توضیح دهید که چرا به این خروجی رسید؟

```
class Classes {
    static class A {
        static int intValue = 0;
        int integerValue = 20;

        A() {
            integerValue = 5;
            printValue();
            print();
        }

        void printCaller() {
            print();
        }

        void printValue() {
            System.out.println("B:" + integerValue);
        }

        void print() {
            System.out.println("A:" + intValue);
        }
    }

    static class B extends A {
        B(int v) {
            intValue = v;
            integerValue = 15;
            printValue();
            print();
        }

        void print() {
            System.out.println("B:" + intValue);
        }

        void printSuper() {
            super.print();
        }

        void printCaller() {
            printValue();
            super.printValue();
        }
    }
}
```

```

        void printValue() {
            System.out.println("B:" + integerValue);
            super.printValue();
        }
    }

    static public class C extends A {
        void printCaller() {
            System.out.println("B:" + integerValue);
        }

        void print() {
            System.out.println("A:" + intValue);
            super.printCaller();
        }
    }
}

class Problem1 {
    public static void incrementValue(Classes.A object) {
        object.intValue++;
        object.integerValue++;
    }

    public static void incrementValue(int firstValue, int secondValue) {
        firstValue++;
        secondValue++;
    }

    public static void main(String[] args) {
        Classes.A a = new Classes.A();
        Classes.B b = new Classes.B(10);
        Classes.A c = b;

        b.print();
        c.print();
        ((Classes.A) b).print();
        b.printSuper();
        a.printCaller();
        b.printCaller();
        c.printCaller();
        incrementValue(a);
        a.printCaller();
        incrementValue(b);
        b.printCaller();
        incrementValue(c);
        c.printCaller();
        incrementValue(b.intValue, b.integerValue);
        b.printCaller();
        c.printCaller();
    }
}

```

سوال ۲

توضیح دهید که هدف از ارث بری در شی گرایی چیست. چه زمان از inheritance و چه زمان از composition استفاده می‌کنیم؟ چگونه می‌توانیم از سازنده پدر را فراخوانی کنیم؟ چگونه می‌توانیم سازنده دیگری از خود کلاس را فراخوانی کنیم؟

هدف ارث بری قابلیت استفاده دوباره از کد است(reuse). اگر رابطه از نوع Is-A باشد از ارث بری و اگر از نوع Has-a باشد از ترکیب استفاده می‌کنیم. با استفاده از Super.

سوال ۳

توضیح دهید که چرا از رابطه‌ها (interface) استفاده می‌کنیم. چه محدودیت‌هایی نسبت به یک کلاس دارند و چرا امکان پیاده‌سازی متدهای آنها داده شده است؟

برای abstraction از اینترفیس استفاده می‌شود. نمی‌توان از آنها object ساخت و همه متدها static و public هستند. برای ارث بری

سوال ۴

کلاس انتزاعی (abstract) چیست و چه زمانی در مدل‌سازی از یک کلاس انتزاعی استفاده می‌کنیم؟ این نوع کلاس چه تفاوتی با رابطه‌ها (interface) دارد؟

از این کلاس‌ها نمی‌توان object تولید کرد و در ارث بری به عنوان کلاس پدر استفاده می‌شوند. در اینترفیس همه متدها abstract هستند و متغیرها final.

سوال ۵

override کردن تابع و متغیر چه تاثیری در عملکرد متدهای کلاس فرزند می‌گذارد؟

باعث می‌شود کلاس فرزند پیاده‌سازی خاص خود را داشته باشد.

چطور می‌توانیم پس از override شدن یک متدهای کلاس فرزند در هر کدام از مکان‌های زیر به نسخه هم نام آن متدهای کلاس پدر دسترسی پیدا کنیم؟

- متدهای داخل کلاس پدر
- متدهای داخل کلاس فرزند
- خارج از دو کلاس

سوال ۶

توضیح دهید که منظور از چندریختی در شی گرایی چیست و چه مزیتی ایجاد می‌کند.

توانایی یک object برای گرفتن انواع مختلف Code reuse.

سوال ۷

چرا از توابع و متدها در زبان برنامه نویسی استفاده می‌کنیم؟ در طراحی برنامه و شکستن آن به توابع و متدهای مختلف چه نکته‌هایی را باید رعایت کرد که خوانایی آن بیشتر شود و پیچیدگی اضافی نداشته باشیم؟

برای استفاده مجدد از آنها

سوال ۸

کلاس درونی (inner class) چه انواعی دارد و هر کدام چه کاربردی در مدل‌سازی و توصیف موجودات دارد؟ چگونه می‌توانیم یک شی از هر نوع ایجاد کنیم؟ در صورت `override` شدن یک متغیر توسط یک کلاس درونی چگونه می‌توان به نسخه `override` شده از کلاس بیرونی دسترسی پیدا کرد؟

سوال ۹

کلمه کلیدی `final` روی هر کدام از موارد زیر چه تاثیری دارد؟

- تابع و متند جلوگیری از overriding
- تعریف کلاس جلوگیری از inheritance
- یک متغیر از نوع شی
- یک متغیر از نوع پایه جلوگیری از تغییر مقدار

سوال ۱۰

کلمه کلیدی `static` روی هر کدام از موارد زیر چه تاثیری دارد؟

- تابع و متند
- تعریف کلاس برای دسترسی به inner نیاز ایجاد شی از نوع outer نیست
- یک متغیر از نوع شی
- یک متغیر از نوع پایه عدم وابستگی به یک object